

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Šorn

**Prototip nizkocenovnega sistema za spremljanje prihodov
na delo z uporabo tehnologije NFC na mobilnih napravah
Android**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2015

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Jure Šorn

**Prototip nizkocenovnega sistema za spremljanje prihodov
na delo z uporabo tehnologije NFC na mobilnih napravah
Android**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Damjan Vavpotič

Ljubljana, 2015

Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljane ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V okviru diplomske naloge razvijte delujoč prototip sistema za spremljanje prihodov na delo, ki bo temeljil na uporabi mobilnih naprav s tehnologijo NFC. Na začetku kratko analizirajte obstoječe sorodne aplikacije ter predstavite njihove prednosti in pomanjkljivosti, nato pa preučite in predstavite ključne tehnologije in rešitve, ki jih je mogoče uporabiti v namen spremljanja prihodov na delo. V nadaljevanju diplomske naloge pripravite delujoč prototip aplikacije, ki bo obsegal tako del na mobilni napravi kot tudi zaledni sistem. Predstavite vse ključne funkcionalnosti, programsko arhitekturo in delovanje prototipa. V zaključku izdelano aplikacijo kritično ovrednotite in podajte smernice za nadaljnje delo.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Jure Šorn, z vpisno številko 63040372, sem avtor diplomskega dela z naslovom:

Prototip nizkocenovnega sistema za spremljanje prihodov na delo z uporabo tehnologije NFC na mobilnih napravah Android

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Damjana Vavpotiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 10. aprila 2015

Podpis avtorja:

Zahvaljujem se vsem, ki so kakorkoli pomagali in pripomogli k uspešno izdelani diplomski nalogi.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Tehnologije in programska oprema.....	3
2.1	Tehnologija NFC	3
2.1.1	Zgodovina.....	3
2.1.2	Delovanje.....	4
2.1.3	NFC v operacijskem sistemu Android	6
2.2	Android	7
2.3	Druge tehnologije in oprema, uporabljena pri razvoju aplikacije.....	7
2.3.1	Eclipse IDE Juno	7
2.3.2	Android SDK.....	7
2.3.3	SQLite.....	8
2.3.4	Jetty.....	8
2.3.5	JSON.....	8
2.3.6	MySQL	8
2.3.7	JBoss EAP	9
2.3.8	Apache Maven.....	9
2.3.9	Sublime Text 2	10
2.3.10	Bootstrap.....	10
2.3.11	Strojna oprema.....	10
Poglavje 3	Sorodni sistemi za evidentiranje prisotnosti	13
3.1	Tušmobil NFC	13
3.2	Adamsoft.....	13

3.3	Četrta pot	15
3.4	Analiza sorodnih rešitev	15
Poglavje 4	Razvoj sistema za spremljanje prihodov na delovno mesto.....	17
4.1	Arhitektura sistema za spremljanje prihodov	18
4.2	Razvoj in implementacija mobilne aplikacije	20
4.2.1	Izdelava podatkovnega modela	20
4.2.2	Zahtevane pravice in minimalna verzija platforme Android	23
4.2.3	Identifikacija uporabnika s tehnologijo NFC	24
4.2.4	Razvoj in implementacija spletnega strežnika na mobilni napravi	26
4.2.5	Razvoj spletne aplikacije	28
4.3	Razvoj strežnika za potrebe sinhronizacije	33
4.3.1	Izdelava podatkovnega modela	33
4.3.2	Spletni servis	34
4.3.3	Sinhronizacija na strani mobilne naprave	35
4.4	Analiza delovanja sistema	36
Poglavje 5	Sklepne ugotovitve	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
NFC	<i>Near Field Communication</i>	komunikacijska tehnologija kratkega dosega
JSON	<i>JavaScript Object Notation</i>	format za prenos podatkov med spletnimi aplikacijami in strežnikom
MySQL	<i>My Structured Query Language</i>	odprtokodni sistem za upravljanje zbirk podatkov
RFID	<i>Radio Frequency Identification</i>	radiofrekvenčna identifikacija
ISO	<i>International Organization for Standardization</i>	Mednarodna organizacija za standardizacijo
CSS	<i>Cascading Style Sheets</i>	predloge, ki določajo vizualno podobo spletne strani
HTML	<i>Hyper Text Markup Language</i>	jezik za označevanje nadbesedila
REST	<i>Representational State Transfer</i>	skupina arhitekturnih omejitev
ECMA	<i>European Computer Manufacturers Association</i>	Evropsko združenje proizvajalcev računalnikov

Povzetek

Mobilne naprave v sodobnih časih vsebujejo vse več tehnologij, med njimi tudi NFC. Diplomsko delo se osredotoča na razvoj sistema, ki podjetjem omogoča evidentiranje prihodov in odhodov zaposlenih z uporabo tehnologije NFC. Ključni del sistema obsega mobilno aplikacijo na operacijskem sistemu Android. Mobilna aplikacija na napravi služi kot terminal za evidentiranje delovnega časa. Za identifikatorje uporablja značke NFC ali mobilne naprave s podporo za tehnologijo NFC. Za urejanje registracij in ostale namembnosti je bila razvita spletna aplikacija, ki je vgrajena v aplikacijo na napravi. Za varno kopiranje in sinhronizacijo podatkov je bil izdelan spletni strežnik. Glavni cilj diplomske naloge je izdelava cenovno ugodne rešitve za evidentiranje delovnega časa zaposlenih, ki bi delovala na eni napravi.

Ključne besede: tehnologija NFC, evidentiranje delovnega časa, mobilna aplikacija, spletna aplikacija, Android.

Abstract

Modern mobile devices comprise many new technologies, including NFC. This thesis focuses on the development of an employee time and attendance system using NFC technology. The main part consists of a mobile application running on the Android platform. A device with this application can be used as a mobile terminal for time and attendance verification. It uses NFC tags and NFC enabled mobile devices for employee identification. Management and monitoring are available through a web interface integrated in the Android application. For synchronization and backup a web server is used. The main objective was to implement a low cost time and attendance system which would work on a single device.

Keywords: NFC technology, recording working time, mobile application, web application, Android.

Poglavje 1 Uvod

Smo v času velikih tehnoloških napredkov. Če smo pred dvajsetimi leti gledali telefon kot nekaj neverjetnega, si sedaj življenja brez njega ne moremo predstavljati. Navadni telefon, ki je skrbel le za klicanje in sporočanje, je sedaj postal pametna naprava, ki pripomore k avtomatizaciji vsakodnevnih opravil. V želji, da bi pametno napravo uporabili na čim več področjih, je treba razviti tudi nove tehnologije. Ena izmed teh tehnologij je brezstična tehnologija *Near Field Communication* (v nadaljevanju NFC), ki omogoča uporabo mobilne naprave za mobilno plačevanje, kupovanje vstopnic, izmenjavo podatkov ipd.

V diplomski nalogi je predstavljena mobilna aplikacija, s katero se uporabnost mobilnih naprav razširi tudi na področje evidentiranja delovnega časa zaposlenih. Aplikacija s pomočjo tehnologije NFC predstavlja cenovno ugodno rešitev, ki vsebuje vse elemente, potrebne za vodenje evidence manjšega podjetja, uvedba aplikacije pa je preprosta in hitra.

Namesto polne denarnice različnih plačilnih kartic, identifikacijskih kartic za spremljanje prihodov na delo, izkaznic za sposojanje knjig in podobno vedno bolj stremimo k uporabi zgolj ene pametne naprave, ki bi navedene kartice in izkaznice učinkovito nadomestila. V ta namen lahko z uporabo sistema za spremljanje prihodov na delo omogočimo uporabo mobilne naprave.

Diplomsko delo je razdeljeno na pet vsebinskih delov. Uvodu sledi poglavje Tehnologije in programska orodja, kjer sta opisana tehnologija NFC ter sistem Android, predstavljeni pa so tudi druge tehnologije in oprema, uporabljene pri razvoju sistema. V tretjem poglavju so predstavljeni na trgu že prisotni sorodni sistemi za evidentiranje prisotnosti na delu. Četrto poglavje opisuje razvoj sistema za spremljanje prihodov na delo. Predstavljeni so arhitektura sistema za spremljanje prihodov, razvoj in implementacija mobilne aplikacije ter razvoj strežnika za potrebe sinhronizacije. V petem poglavju je delo strnjeno v sklepne ugotovitve.

Na koncu diplomske naloge so navedeni literatura in viri, ki so bili osnova za pisanje diplomskega dela.

Poglavje 2 Tehnologije in programska oprema

V tem poglavju diplomskega dela bodo podrobneje predstavljene naslednje tehnologije in programska oprema:

- tehnologija NFC,
- operacijski sistem Android,
- druge tehnologije in programska oprema.

2.1 Tehnologija NFC

Komunikacijska tehnologija kratkega dosega, krajše NFC (angl. *Near Field Communication*), je visokofrekvenčna komunikacijska tehnologija, ki omogoča izmenjavo podatkov na razdalji do 10 cm in se uporablja predvsem pri mobilnih telefonih. Trije glavni uporabniški primeri so [1]:

- kot kartica; naprava NFC deluje kot obstoječa brezkontaktna kartica,
- kot čitalec; naprava NFC deluje v aktivnem načinu in bere pasivne kartice,
- način P2P; dve napravi NFC komunicirata med seboj in si izmenjujeta informacije.

Pozitivni lastnosti tehnologije NFC sta nizka poraba električne energije in kratek vzpostavitevni čas.

2.1.1 Zgodovina

Tehnologija NFC predstavlja nadgradnjo tehnologije RFID (radiofrekvenčna indentifikacija ali angleško *Radio Frequency Identification*). RFID je tehnologija za enosmerni prenos podatkov med čitalcem in elektronsko oznako z namenom identifikacije. Zaradi cenovne ugodnosti in preprostosti je uporaba te tehnologije priljubljena v proizvodnji ter skladiščenju, pri plačevanju in na drugih področjih. NFC v primerjavi z RFID omogoča dvosmerno

izmenjavo podatkov, ima krajši doseg, kar pomeni večjo varnost, ter se povezuje tudi z drugimi obstoječimi tehnologijami, kot so Wi-Fi in Bluetooth.

Prvič je bila tehnologija NFC uradno predstavljena leta 2002, ko sta podjetji Philips in Sony predstavili brezkontaktno povezavo, ki sta jo kot standard z oznako ISO/IEC 18092 decembra 2003 sprejeli Mednarodna organizacija za standardizacijo ISO ter Mednarodna elektrotehnična komisija IEC [3].

V letu 2004 so Nokia, Sony in Philips skupaj ustanovili NFC Forum. Namen združenja je bil povečati popularnost, enostavnost in varnost ter zagotavljati združljivost med napravami in storitvami NFC. Leta 2006 so bile izdane prve specifikacije za značke NFC [2], ki vsebujejo informacijo, le-to pa kompatibilna naprava, opremljena s tehnologijo NFC, pridobi z približanjem. Ta informacija je lahko poljubne narave in lahko sproži različne akcije. Telefon Nokia 6131, razvit v februarja 2006, je bil prvi pametni telefon z NFC.

Od leta 2006 do danes je bilo izdanih več specifikacij, tehnologija pa podpira tudi plačilne metode ter izmenjavo datotek in povezav. Leta 2010 je na prodajne police prišel prvi pametni telefon z operacijskim sistemom Android, Samsung Nexus S [2].

2.1.2 Delovanje

NFC deluje na frekvenci 13,56 MHz s pasovno širino 2 MHz na osnovi magnetne indukcije in za komunikacijo porablja manj kot 15 mA [6]. Bralnik oddaja šibek električni tok, ki ustvari magnetno polje med napravama, to magnetno polje druga naprava sprejme in ga spremeni nazaj v električne impulze ter nato v podatke. Standardno so podprte hitrosti 106, 212, 424 in 848 kbit/s.

Poznamo tri različne načine delovanja (Slika 2.1). Prvi način je posnemanje kartice (angl. *Card Emulation*), pri katerem aktivna naprava NFC posnema delovanje pametne kartice. Napravo lahko uporabimo za plačevanje vstopnic in vozovnic. Drugi način je način vsak z vsakim (angl. *Peer to Peer Mode*), ki omogoča komunikacijo med dvema aktivnima napravama NFC. Zaradi hitrosti se ta način uporablja za ustvarjanje sekundarne povezave z Bluetoothom ter izmenjevanje kontaktov in podobnih podatkov. Za izmenjavo podatkov obstajata dva protokola; NFCIP-1 (angl. *NFC Interface and Protocol-1*), ki je narejen po vzorcu pobudnik–cilj, v katerem sta obe napravi definirani pred začetkom komunikacije, ter LLCP (angl. *Logical Link Control Protocol*), kjer obe aktivni napravi delujeta tako, da med oddajanjem prve druga posluša in začne oddajati podatke, ko prva zaključi. Tretji način je branje/pisanje in je namenjen komunikaciji med aktivno napravo NFC (na primer mobilna naprava) in značko NFC. V primeru branja mobilna naprava prebere podatke, shranjene na

znački, v pisalnem načinu delovanja pa se obstoječi podatki izbrišejo in zapišejo se novi. Primeri za ta način delovanja so branje urnikov, branje pametnih plakatov in prevzemanje kuponov.



Slika 2.1: Načini delovanja naprav NFC.

Podatki, ki se prenašajo s pomočjo tehnologije NFC, so zapisani v osnovnem formatu sporočil NDEF (angl. *NFC Data Exchange Format*). Vsako sporočilo formata NDEF (angl. *NDEF message*) je lahko sestavljeno iz enega ali več zapisov (angl. *NDEF record*) [4]. V vsakem zapisu so podatki in glava zapisa, ki vsebuje tip, identifikator ter dolžino podatkov. Za standardne tipe zapisa, ki se uporabljajo v aplikacijah NFC Foruma, skrbi predpisan format RTD (angl. *Record Type Definition*). Specifikacija RTD novim aplikacijam določa način za učinkovito opredelitev oblik zapisa in daje uporabnikom možnost, da ustvarijo lastne aplikacije, ki temeljijo na specifikacijah NFC Foruma [5]. RTD predpisuje formate zapisa za shranjevanje besedila, spletnih naslovov, zahtev za izvajanje določenih akcij na napravi NFC, podpisov ter dodatnih informacij, ki jih pripnemo pametnemu plakatu.

Pomembno vlogo pri komunikaciji tehnologije NFC imajo tudi značke NFC. Tipična značka NFC je sestavljena iz mikročipa, ki hrani podatke, in je priklopljena na radijsko anteno. NFC Forum je objavil štiri tipe značk NFC (tabela 2.1), ki so kompatibilni z napravami NFC. Razlikujejo se glede na velikost spomina, hitrost prenosa, obliko in izvor energije.

Parametri	Tip 1	Tip 2	Tip 3	Tip 4
Standard	ISO14443	ISO14443	Sony FeliCa	ISO14443
Velikost spomina	do 1 KB	do 2 KB	do 1 MB	do 64 KB
Hitrost prenosa	106 kbit/s	106 kbit/s	212 kbit/s	106–424 kbit/s
Cena	nizka	nizka	visoka	srednja

Tabela 2.1: Tipi značk NFC [3].

2.1.3 NFC v operacijskem sistemu Android

Android je z verzijo 2.3 (*API level 9*) vpeljal podporo za NFC [3]. Uradne knjižnice sistema Android omogočajo izmenjavo podatkov NFC v obliki paketov NDEF (angl. *NFC Data Exchange Format*) med dvema napravama NFC z uporabo Androidove storitve Android Beam ali med napravo in značko NFC.

Z branjem podatkov NDEF z značke se ukvarja sistem za odpremo značk (angl. *tag dispatch system*). Naloga tega sistema je analizirati zaznane značke NFC in ugotoviti, kateri uporabniški aplikaciji podatke posredovati. Sistem deluje v treh korakih [7]:

- branje značke NFC in ugotavljanje tipa zapisanih podatkov,
- razvrstitev vsebine značke v razred za nadaljnjo obdelavo v aplikaciji,
- zagon aplikacije in posredovanje informacij.

Če je uporabniških aplikacij, registriranih za obdelavo enakega tipa podatkov NFC, več, uporabnik sam izbere najprimernejšo. Če takšne aplikacije ni, sistem poizkuša najti aplikacijo, ki je registrirana za obdelavo širše kategorije podatkov in dostopa do najbolj splošnih aplikacij, ki berejo značke NFC. Če tudi takih aplikacij ni, ne naredi ničesar.

Če želimo razviti aplikacijo, ki bo obdelala določen tip značk, moramo to deklarirati v konfiguracijski datoteki `AndroidManifest.xml`, v kateri dodajamo pravice do uporabe tehnologije NFC.

2.2 Android

Android je mobilni operacijski sistem, ki je osnovan na Linuxovem jedru [6]. Operacijski sistem je bil sprva razvit predvsem za pametne mobilne naprave in tablice, sedaj pa se uporablja tudi v pametnih televizijah, avtih ter urah. Androidova odprtokodna programska oprema je v lasti spletnega giganta Google.

Aplikacije za sistem Android vedno bolj izkoriščajo različne funkcionalnosti telefona. Ker so te funkcionalnosti omejene v vsaki verziji Androida, se vzpostavlja konstantna potreba po razvoju novih. Prodaja aplikacij za sistem Android in iger se izvaja s pomočjo spletne trgovine Google Play. Tam lahko ponudnik aplikacije sam določa pogoje prodaje izdelka. Lahko ga trži po celem svetu ali se osredotoči na določeno območje, prav tako lahko postavi ceno po svoji izbiri ali pa aplikacijo uporabnikom ponudi brezplačno.

Vsaka aplikacija za sistem Android je sestavljena iz vsaj ene izmed štirih komponent: aktivnost (angl. *activity*), storitev (angl. *service*), upravitelj vsebine (angl. *content provider*) in prejemnik sporočil (angl. *broadcast receiver*) [3]. Aktivnost predstavlja uporabniški vmesnik aplikacije, s katerim je mogoče izvajati različne akcije. Storitve je namenjena izvajanju procesa v ozadju. Upravitelj vsebine je komponenta, s katero se preko vmesnika definira dostop do podatkov aplikacije. Prejemnik sporočil pa je komponenta, ki prejema in reagira na sporočila, ki so poslana iz drugih aplikacij ali samega sistema.

2.3 Druge tehnologije in oprema, uporabljene pri razvoju aplikacije

V tem poglavju so predstavljene tehnologije in oprema, ki so bile uporabljene pri razvoju celotnega sistema, od mobilne aplikacije do tehnologij na strani strežnika.

2.3.1 Eclipse IDE Juno

Eclipse je razvojno okolje za razvijanje različnih aplikacij. Izdelano je v okviru skupnosti za posameznike in organizacije, ki želijo sodelovati pri razvoju komercialno prijazne odprtokodne programske opreme. S pomočjo vtičnikov (angl. *plug-in*) razvijalcem omogoča uporabo različnih programskih jezikov, kot so C++, Perl, PHP, Python, R, Scala in drugi.

2.3.2 Android SDK

Android SDK (angl. *software development kit*) je programski paket, ki omogoča razvoj aplikacij za platformo Android. Android SDK vsebuje primere projektov s programsko kodo,

razvojna orodja, emulatorje in knjižnice, potrebne za izdelavo aplikacij za sistem Android [6]. Te aplikacije so zapakirane v formatih s končnico .apk in shranjene v zaščitenem /data/app direktoriju operacijskega sistema Android.

2.3.3 SQLite

SQLite je odprtokodna podatkovna baza, ki za svoje delovanje ne potrebuje strežnika, ampak lahko piše neposredno na trdi disk [10]. Vsa podatkovna baza je shranjena v eni sami datoteki, kar je tudi vzrok, da je SQLite najpogostejše uporabljena podatkovna baza v operacijskem sistemu Android. Baza SQLite je zelo previdno testirana in vsaka nova verzija velja za zelo zanesljivo. Prav tako ni občutljiva na razliko med 32- in 64-bitnim sistemom ter arhitekturo debelega in tankega konca.

2.3.4 Jetty

Jetty je javanski spletni strežnik in skupek javanskih servletov, ki omogočajo posredovanje dinamične in statične vsebine. Je brezplačen odprtokodni projekt, ki spada pod fundacijo Eclipse. Uporabljajo ga Apache ActiveMQ, Alfresco, Apache Geronimo, Apache Maven, Apache Spark, Google App Engine, Eclipse, FUSE in drugi. Jetty je tudi strežnik v odprtokodnih projektih, kot so Lift, Eucalyptus, Red5, Hadoop in I2P. Razvijati se je začel v letu 1995 in od takrat je bilo razvitih že 9 verzij. Podpira tehnologije AJP, JASPI, JMX, JNDI, OSGi in WebSocket [11]. Za razvoj našega sistema je bila uporabljena verzija 7.x, saj deluje na verziji Jave 1.5, ki je naložena na testnem mobilnem telefonu.

2.3.5 JSON

JSON (angl. *JavaScript Object Notation*) je standardni format za izmenjavo človeško berljivih podatkov. Prvotno je bil razvit za izmenjavo podatkov med strežnikom in spletnimi aplikacijami kot alternativa za XML. Čeprav izvira iz skriptnega jezika JavaScript, je JSON jezikovno neodvisen podatkovni format. Koda za generiranje in razčlenjevanje podatkov JSON je na voljo v številnih programskih jezikih. Format JSON je trenutno opisan z dvema standardoma: RFC 7159 in ECMA-404 [12]. Standard ECMA je minimalen in opisuje le dovoljeno slovnično skladnjo, medtem ko RFC opisuje tudi semantične in varnostne vidike. Uradni tip MIME za JSON je application/json.

2.3.6 MySQL

MySQL je odprtokodni sistem za upravljanje s podatkovnimi bazami, ki za delo s podatki uporablja jezik SQL. Za njegove razvoj, distribucijo in vzdrževanje skrbi podjetje Oracle [13]. MySQL je sprogramiran v programskih jezikih C in C++ ter deluje na različnih operacijskih sistemih. Za razvoj sistema za shranjevanje podatkov na centralnem strežniku smo si pomagali s spletnim vmesnikom phpMyAdmin, ki je poskrbel za dostop do podatkovne baze. Vmesnik phpMyAdmin, ki temelji na jeziku PHP, omogoča pregled podatkov in upravljanje MySQL preko spletnega grafičnega vmesnika. Za poizvedovanje po podatkih v podatkovni bazi se uporablja poizvedovalni jezik SQL.

2.3.7 JBoss EAP

JBoss EAP (angl. *JBoss Enterprise Application Platform*) je aplikacijska strežniška platforma, ki omogoča gostovanje javanskih spletnih aplikacij in storitev. Platforma Jboss EAP je zgrajena na osnovi inovativne arhitekture JBoss Microcontainer in nudi izboljšano navzkrižno nalaganje in delovanje ves čas uporabe ter fleksibilnost za širok nabor programskih in komponentnih modulov, ki vključujejo Java EE, POJOs, OSGi in Spring [14]. Platforma JBoss je v lasti podjetja Red Hat in je v celoti brezplačna. V okviru Jboss EAP je Red Hat razvil tudi JBoss Developer Studio, ki pripomore k lažjemu razvijanju programske kode. Jboss Developer Studio je grafični vmesnik, ki temelji na Eclipsu. Poleg standardnih Eclipsevih funkcij Developer Studio nudi tudi direkten dostop do strežniških nastavitev, zagon in brisanje spletnih aplikacij ter ostale ključne funkcije. JBoss EAP je tesno povezan z repozitorijem Maven, ki je opisan v nadaljevanju.

2.3.8 Apache Maven

Apache Maven je distribuirano orodje, uporabljeno v javanskih aplikacijah za razvijanje, kreiranje in upravljanje programskih projektov. Maven uporablja standardno konfiguracijsko datoteko POM (angl. *Project Object Model*), ki definira celoten projekt [15]. V datoteki POM (pom.xml) so opisi izvajanja projekta ter vseh odvisnih knjižnic, ki so potrebne za izvajanje projekta. Primer datoteke pom.xml je prikazan na Sliki 2.1. Vse uporabljene javanske knjižnice se prenese s spletnih repozitorijev Maven, shranjujejo pa se v interno shrambo na trdem disku. JBoss EAP ima svoj repozitorij Maven, ki vsebuje vse potrebne knjižnice za zagon in izvajanje spletnih aplikacij na strežniku.

```
<project>

  <modelVersion>4.0.0</modelVersion>
  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0</version>

  <dependencies>
    <dependency>
      <groupId>org.json</groupId>
      <artifactId>json</artifactId>
      <version>20140107</version>
    </dependency>
  </dependencies>

</project>
```

Slika 2.1: Primer datoteke pom.xml, ki v projekt vključi knjižnico za delo z objekti JSON.

2.3.9 Sublime Text 2

Sublime Text 2 je urejevalnik za programsko kodo, označevanje in prozo. Uporablja zelo enostaven uporabniški vmesnik in posebne funkcije, ki jih je mogoče nadgraditi z različnimi vtičniki. Le-ti so na voljo brezplačno in omogočajo različne funkcionalnosti za različne programske jezike [24].

2.3.10 Bootstrap

Bootstrap je odprtokodna zbirka sodobnih orodij za izdelovanje spletnih strani in spletnih aplikacij [25]. Vsebuje številne lepo oblikovane komponente HTML, kot so tabele, obrazci, ikone, gumbi, zaslonski namigi (angl. *tooltips*), spustni meniji (angl. *drop-down menus*) in še veliko drugih.

2.3.11 Strojna oprema

Za testiranje sistema sta bili uporabljeni napravi:

- Samsung Galaxy S3 in
- Samsung Galaxy S4.

Samsung Galaxy S3

Specifikacije naprave [16]:

- Zaslon: 4,8 " (12,2 cm),
- Resolucija: 720 x 1280 slikovnih točk,
- Procesor: štirijedrni 1,4 GHz,
- Delovni spomin: 1024 MB RAM,
- NFC: da.

Samsung Galaxy S4

Specifikacije naprave [17]:

- Zaslon: 5,0 " (12,7 cm),
- Resolucija: 1080 x 1920 slikovnih točk,
- Procesor: štirijedrni 1,6 GHz,
- Delovni spomin: 2048 MB RAM,
- NFC: da.

Poglavje 3 Sorodni sistemi za evidentiranje prisotnosti

Podjetja lahko za spremljanje prihodov in odhodov svojih zaposlenih izbirajo med različnimi sistemi. Ti sistemi se razlikujejo glede na vrsto terminala, ki ga uporabljajo, in način tehnologije. Nekatera podjetja so na tržišču že ponudila sisteme, ki uporabljajo tehnologijo NFC. Ta podjetja in njihovi produkti bodo predstavljeni v nadaljevanju.

3.1 Tušmobil NFC

Podjetje Tušmobil je s 1. 5. 2014 uredil splošne pogoje glede prodajanja sistema za vodenje evidence prisotnosti z uporabo telefonov NFC [18]. Ta sistem omogoča individualno evidentiranje prisotnosti med delovnim časom in združuje aplikacijo za programiranje oznak NFC, spletno aplikacijo za urejanje seznama zaposlenih in pregledovanje zabeleženih časov (na Sliki 3.1) ter spletno aplikacijo za obdelovanje zabeleženih časov.

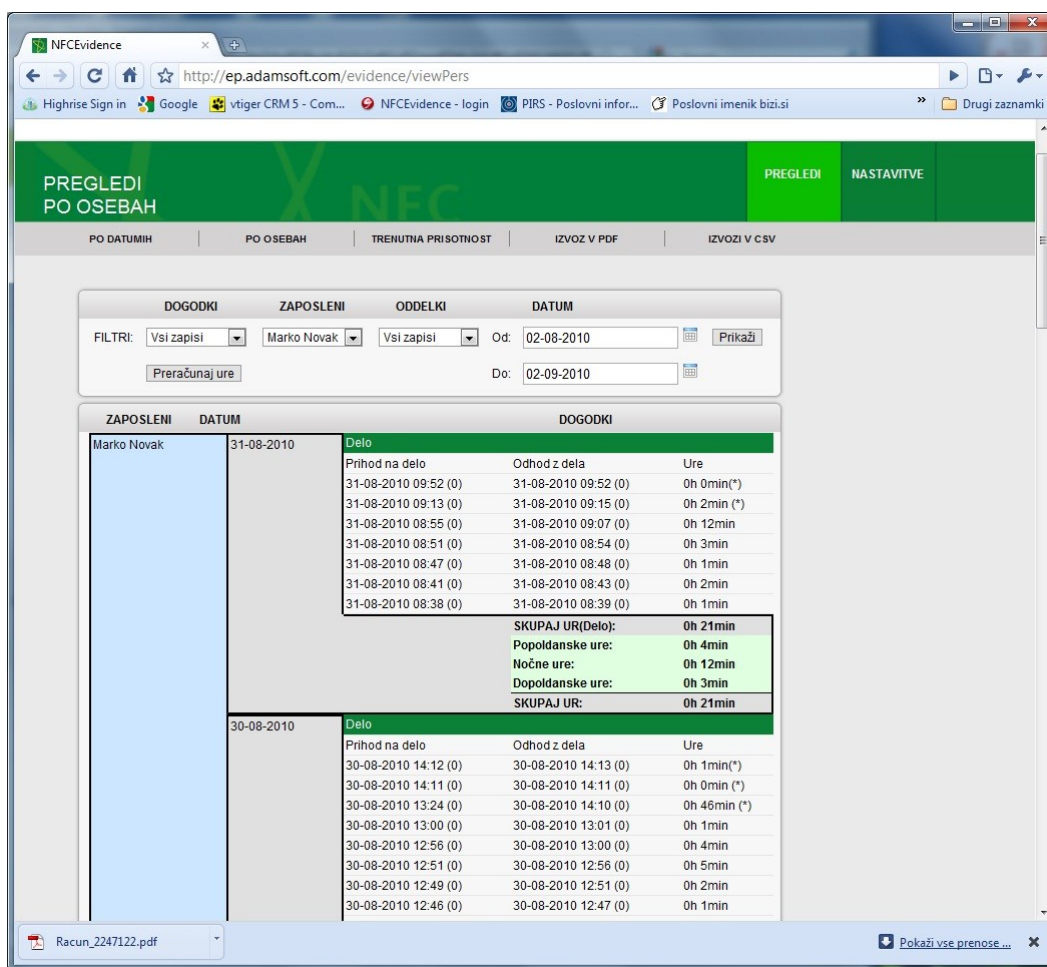
The screenshot shows the Tušmobil web application interface. At the top, there is a navigation bar with tabs: 'PREGLEDI', 'NASTAVITVE', and 'ODJAVA'. Below this is a dark header with the 'tušmobil' logo and the tagline 'ustvarjamo bližino'. To the right of the logo are several menu items: 'PO DATUMIH', 'PO OSEBAH', 'TRENUTNA PRISOTNOST', 'IZVOZ V PDF', and 'IZVOZ V CSV'. The main content area features a table with columns: 'DOGODKI', 'ZAPOSLENI', 'ODDELKI', and 'DATUM'. Above the table, there are filter options labeled 'FILTRI' with dropdown menus for 'Vsi zapisi' under each column. To the right of the filters are date selection fields: 'Od 20-5-2014' and 'Do 20-5-2014', along with buttons 'PRERAČUNAJ URE' and 'PRIKAŽI'. Below the table, there are additional labels: 'ZAPOSLENI', 'DATUM', and 'DOGODKI'.

Slika 3.1: Spletna aplikacija za obdelovanje zabeleženih časov podjetja Tušmobil.

3.2 Adamsoft

Podjetje Adamsoft ponuja rešitev za evidentiranje zaposlenih s pomočjo tehnologije NFC, in sicer sistem Time & Attendance. NFC Time & Attendance je dvonivojska rešitev. Za delovanje tega sistema je podjetje Adamsoft razvilo mobilno aplikacijo, ki se usklajuje z zalednim sistemom (Slika 3.2), dostopnim preko spletnega vmesnika. Mobilna aplikacija služi za odčitavanje kartic NFC in pošiljanje podatkov na strežnik. Primer uporabe sistema za končnega uporabnika (zaposlenega v podjetju) predvideva dva osnovna scenarija [19]:

- Zaposleni pride na delovno mesto in se s telefonom, ki omogoča NFC, dotakne določenega mesta (značke NFC) – na telefonu se zažene aplikacija, kjer so predhodno nastavljeni poljubni dogodki (prijod, službeni prihod, malica) – uporabnik izbere ustrezen dogodek – ta se takoj prenese v uporabnikovo bazo na strežniku (nahaja se pri MNO-ju).
- Zaposleni pride na delovno mesto in se s kartico NFC dotakne telefona, ki omogoča NFC in služi kot terminal za evidenco delovnega časa. Nadaljnji postopek je enak kot pri zgornji točki.



Slika 3.2: Spletni sistem za pregled evidentiranj podjetja Adamsoft (<http://www.adamsoft.si/f/img/Image/tina.jpg>).

diplomska naloga predstavlja razširitev informacijskega sistema za registracijo delovnega časa podjetja Špica International na uporabo tehnologije NFC in ocenjuje ustreznost te tehnologije. Osnova za delovanje te rešitve je sistem Time & Space. Vse omenjene rešitve temeljijo na terminalu za evidentiranje prisotnosti, ki je v našem primeru mobilna naprava, in brezkontaktnih karticah, ki predstavljajo identifikatorje zaposlenih. Vsa evidenca zaposlenih se hrani na spletnem strežniku ponudnika, ki za določeno ceno nudi vodenje, upravljanje in vzdrževanje spletnih storitev. Te cene so različne ter odvisne od ponudnikov samih in storitev, ki jih podjetje želi.

Rešitev, ki jo predstavlja ta diplomska naloga, temelji na cenovno ugodni rešitvi in samozadostnosti. Ob namestitvi naše aplikacije na mobilno napravo s podporo tehnologije NFC le-ta postane celovit registracijski informacijski sistem, ki nudi evidentiranje s pomočjo kartic NFC in mobilnih naprav ter vzpostavitev spletnega strežnika, ki omogoča urejanje evidence delovnega časa. Prednost implementacije strežnika na mobilni napravi je ravno v tem, da mobilna aplikacija postane v celoti samozadostna ter da lahko s preprosto razširitvijo dodamo skoraj vse funkcije mobilne naprave in jih implementiramo v spletni aplikaciji. Vzpostavitev in vzdrževanje strežnika predstavljata zelo majhen strošek, saj je mobilna naprava z vidika porabe električne energije zelo varčna. Če podjetje nima tovrstne mobilne naprave, strošek vzpostavitve sistema za podjetja predstavlja le nakup mobilne naprave s podporo tehnologije NFC.

Ponudnikov in povpraševalcev za sisteme evidentiranja je veliko. Vsak dan se odpirajo nova podjetja, ki potrebujejo vodenje evidence zaposlenih. Njihove želje glede načina evidentiranja so različne. Odločili smo se, da na trgu ponudimo novo rešitev za podjetja z majhnim številom zaposlenih, ki jim je do sedaj vodenje evidence zaradi cenovnih in lokacijskih težav predstavljalo problem ali pa so uporabljali še stare načine evidentiranja, ki bi jih radi nadgradili.

Poglavje 4 Razvoj sistema za spremljanje prihodov na delovno mesto

V nadaljevanju bosta predstavljena načrtovanje in implementacija sistema, ki demonstrira uporabo tehnologije NFC. Sistem sestavljata mobilna aplikacija in strežniški del, ki skrbi za varno shranjevanje podatkov. Aplikacija je poimenovana NFC Signer in bo lahko prepoznana po logotipu na Sliki 4.1. Aplikacija bo vsebovala vse lastnosti terminala za evidentiranje prihodov zaposlenih s pomočjo tehnologije NFC. Ker je naš ključni namen znižati stroške podjetij ob uvedbi sistema za spremljanje prihodov, smo se odločili v aplikacijo vgraditi še strežniški del. Zaradi možne zlorabe, okvare naprave ali drugih težav je podatke vedno pametno varnostno kopirati. V ta namen smo dodali centralno bazo podatkov, ki se sinhronizira z bazo na mobilni napravi.

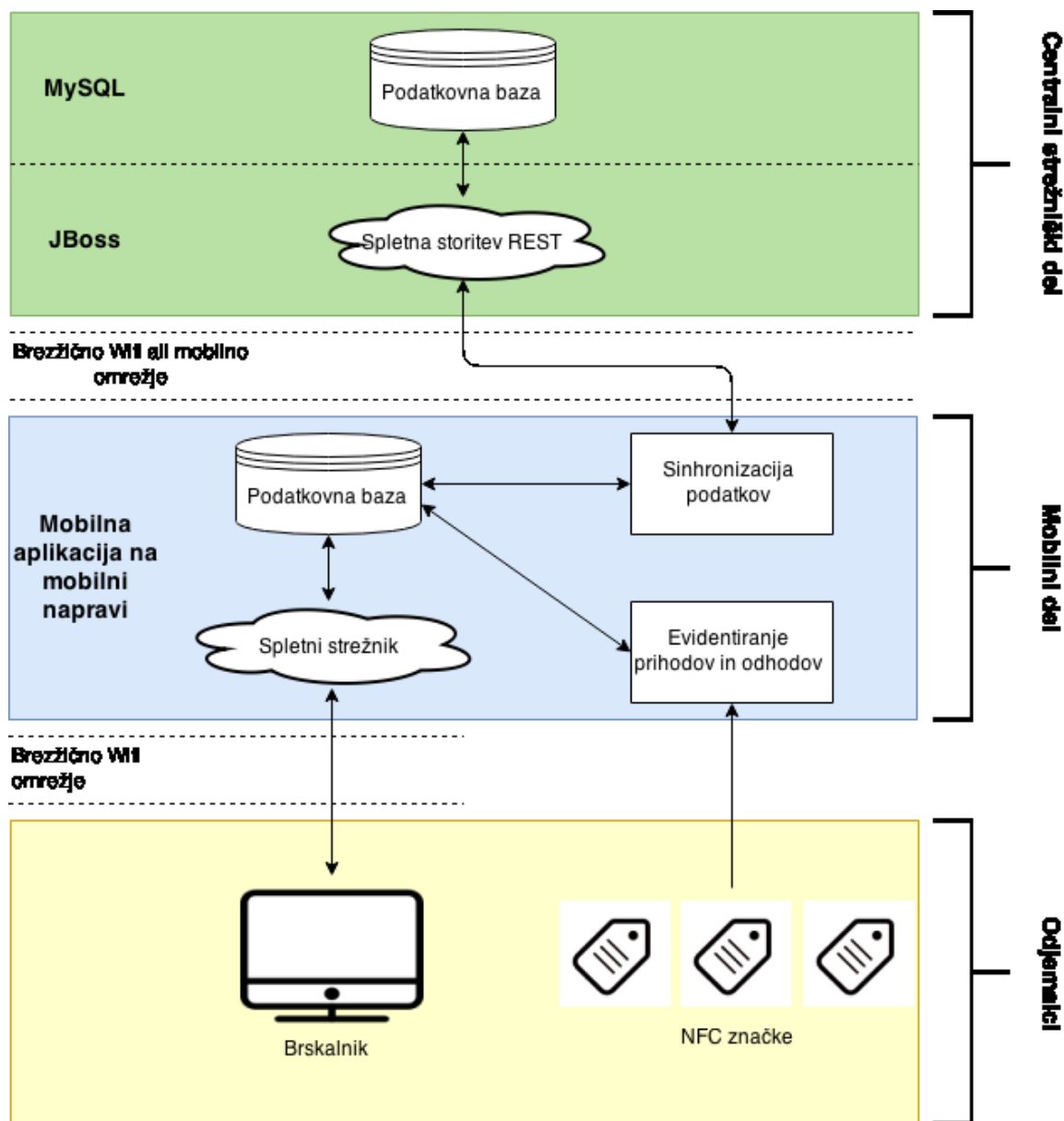
Sistem za evidentiranje prihodov je za nova podjetja stroškovno ugoden začetek. Uslužbenci bi se lahko registrirali s pomočjo svojih mestnih kartic Urbana ali pa z uporabo svoje mobilne naprave, če ta podpira tehnologijo NFC. Strošek bi za podjetje tako predstavljala le mobilna naprava NFC ter strežnik za varnostno kopiranje podatkov.



Slika 4.1: Logotip aplikacije NFC Signer. Sivi krog simbolizira uro ali čas, simbol znotraj njega pa mobilno napravo s tehnologijo NFC.

4.1 Arhitektura sistema za spremljanje prihodov

Na Sliki 4.2 je predstavljena arhitektura celotnega sistema. V grobem se sistem deli na mobilno aplikacijo in strežniški del, ki skrbi le za sinhronizacijo podatkov. Mobilna naprava z nameščeno mobilno aplikacijo NFC Signer ima vlogo terminala za evidentiranje prihodov uslužbencev ter glavnega strežnika, ki posreduje spletne strani.



Slika 4.2: Arhitektura sistema.

Sistem pozna dve vrsti uporabnikov; uporabnike s skrbniškimi pravicami in navadne uporabnike. Na konkretnem primeru so uporabniki s skrbniškimi pravicami direktorji in

uslužbenci z večjimi pravicami, navadni uporabniki pa so ostali uslužbenci. Uporabniki, ki imajo skrbniške pravice, lahko s pomočjo glavnega spletnega strežnika dostopajo do seznama vseh uslužbencev ter dodajajo, urejajo in pregledujejo vse prihode zaposlenih. Navadni uporabniki imajo pravice za pregledovanje svojih prihodov in odhodov ter spreminjanje svojega profila.

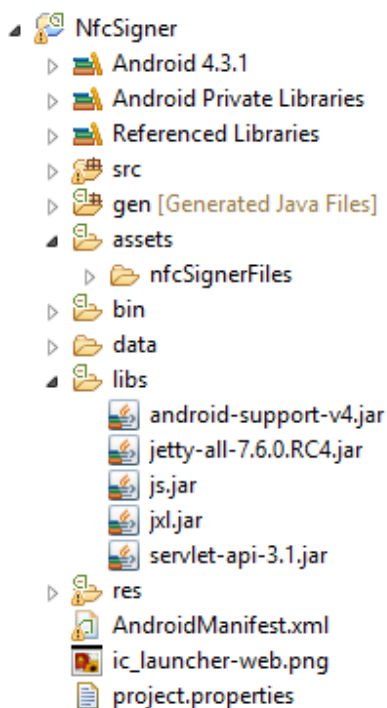
Mobilna naprava je v našem primeru lahko katerikoli pametni telefon z operacijskim sistemom Android od verzije 4.3 (Jelly Bean) dalje.

V nadaljevanju bomo podrobneje predstavili mobilni in strežniški del sistema.

4.2 Razvoj in implementacija mobilne aplikacije

Mobilno aplikacijo sestavljata aplikativni del in podatkovna baza. Aplikativni del skrbi za grafični vmesnik, identifikacijo zaposlenega, sinhronizacijo podatkov ter spletni vmesnik. Podatkovna baza je zadolžena za shranjevanje podatkov o vseh uporabnikih in dogodkih.

Struktura projekta iz razvojnega okolja Eclipse je prikazana na Sliki 4.3.



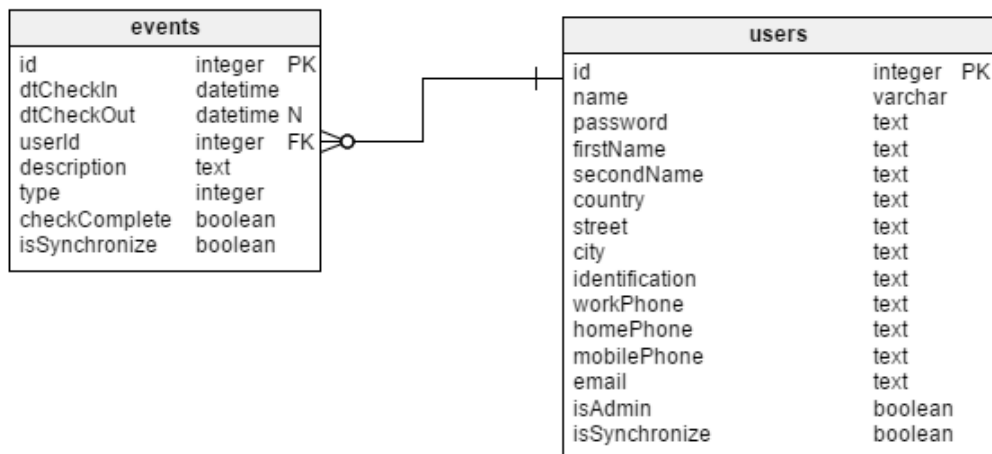
Slika 4.3: Struktura projekta NfcSigner.

Android ima za vse datoteke določeno razporeditev. V mapi src je izvorna koda v javanskem jeziku. Mapa res vsebuje definicije postavitve grafičnega vmesnika, besedila, slike. V mapi lib so vse dodatne knjižnice, ki so potrebne za pravilno delovanje aplikacije. V mapi assets je mapa nfcSignerFiles, ki vsebuje datoteke za spletni vmesnik.

4.2.1 Izdelava podatkovnega modela

Vsi moduli in funkcije se začnejo z izdelavo podatkovnega modela, saj brez podatkov ne vemo, kateri uporabnik se je identificiral. Na podlagi zahtev za identifikacijo s tehnologijo NFC in evidentiranje prisotnosti smo definirali, katere entitete mora aplikacija vsebovati.

Na mobilni napravi se v lokalni bazi SQLite hranijo vsi podatki. Obenem imajo možnost samodejnega kopiranja v centralno bazo podatkov. Podatkovna baza SQLite hrani dve entiteti: **users** in **events**, prikazani na Sliki 4.4. Prva skrbi za shrambo vseh uporabnikov, v drugi pa so podatki o dogodkih. Vsak uporabnik ima lahko nič ali več dogodkov v entiteti **events**, medtem ko en vnos entitete **events** pripada samo eni osebi.



Slika 4.4: Shema podatkovne baze, uporabljena na mobilni napravi. Poleg imena atributa je dodan še tip zapisa podatka.

V nadaljevanju sledi opis atributov posameznih entitet, najprej entitete **events** (Tabela 4.1), nato entitete **users** (Tabela 4.2).

Polje	Opis polja
id	Šifra dogodka. Umetni enolični identifikator.
dtCheckIn	Čas začetka dogodka. Čas in datum začetka dogodka.
dtCheckOut	Čas konca dogodka. Čas in datum konca dogodka.
userId	Uporabniška šifra. Predstavlja uporabnika, na katerega je vezan ta dogodek.
description	Dodaten opis dogodka. Dogodek lahko v primeru dodatnih pojasnil tudi opišemo.
type	Tip dogodka. Vsak dogodek lahko uvrstimo v enega izmed štirih tipov dogodkov: redno v službi, na bolniški, v službeni odsotnosti, na dopustu.

checkComplete	Popolnost dogodka. Dogodek je popoln v primeru, ko sta podana tako začetek kot konec dogodka.
isSynchronize	Sinhronizacija dogodka. Polje se uporablja za sinhronizacijske namene. Pove nam, ali je bil nek dogodek že sinhroniziran.

Tabela 4.1: Opis strukture podatkov entitete **events**.

Polje	Opis polja
id	Šifra uporabnika. Identifikator uporabnika.
name	Uporabniško ime. Uporabniško ime uporabnik potrebuje za vstop v spletno aplikacijo ter za izvajanje različnih akcij v mobilni aplikaciji.
password	Geslo uporabnika. Uporabniško geslo za vstop v spletno aplikacijo ter za izvajanje različnih akcij v mobilni aplikaciji.
firstName	Ime uporabnika.
secondName	Priimek uporabnika.
country	Državljanstvo uporabnika.
street	Ulica in hišna številka stalnega prebivališča uporabnika.
city	Mesto stalnega prebivališča uporabnika.
identification	Identifikacija NFC. Predstavlja identifikator značke NFC.
mobilePhone	Mobilna telefonska številka.
homePhone	Domača telefonska številka.
workPhone	Službena telefonska številka.
email	Elektronska pošta uporabnika.

isAdmin.	Administrator. To polje nam pove, ali ima uporabnik skrbniške pravice.
isSynchronize	Sinhronizacija dogodka. Polje se uporablja za sinhronizacijske namene. Pove nam, ali so bile spremembe uporabnika že sinhronizirane.

Tabela 4.2: Opis strukture podatkov entitete **users**.

4.2.2 Zahtevane pravice in minimalna verzija platforme Android

Aplikacija je zasnovana s strežniškim delom sistema in tehnologijo NFC. Za delo s slednjo je treba mobilni aplikaciji dodeliti pravice za dostop do modula NFC. Vse pravice in zahteve se nahajajo v konfiguracijski datoteki AndroidManifest.xml (Slika 4.5).

```
<uses-sdk
    android:minSdkVersion="18"
    android:targetSdkVersion="22" />

<uses-permission android:name="android.permission.NFC" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-feature android:name="android.hardware.nfc" />
```

Slika 4.5: Prikaz najnižje sprejemljive različice API-ja naprave ter dodelitev pravic do uporabe interneta, tehnologije NFC in uporabe pomnilnika naprave v datoteki AndroidManifest.xml.

Za najboljšo izkušnjo uporabe mobilne aplikacije je potrebna internetna povezava, na napravi pa mora biti omogočena tudi tehnologija NFC. Dodatne pravice za uporabo tehnologije NFC niso potrebne, saj slednjo aplikacija uporablja le za branje. Dodana je tudi pravica za pisanje na pomnilnik mobilne naprave, saj podatke spletne aplikacije prekopiramo na pomnilnik. Zaradi uporabe različnih knjižnic je aplikacija omejena na naprave z API-jem, novejšim od različice 18 (Jelly Bean).

Ob zagonu aplikacija preveri dostopnost internetne povezave v primeru vključene možnosti sinhronizacije podatkov. Če na mobilni napravi tehnologija NFC ni omogočena, uporabniku z dialoškim oknom ponudi možnost dostopa do nastavitev za vklop te tehnologije. V primeru, da naprava ne podpira tehnologije NFC, uporabnika na to opozori in konča z izvajanjem mobilne aplikacije.

4.2.3 Identifikacija uporabnika s tehnologijo NFC

Značke NFC odčitavata razreda `UserActivity` in `MainActivity`. Prva aktivnost ima nalogo urejanja in dodajanja uporabnikov ter njihovih identifikatorjev NFC, druga aktivnost pa skrbi za evidentiranje prihodov in odhodov.

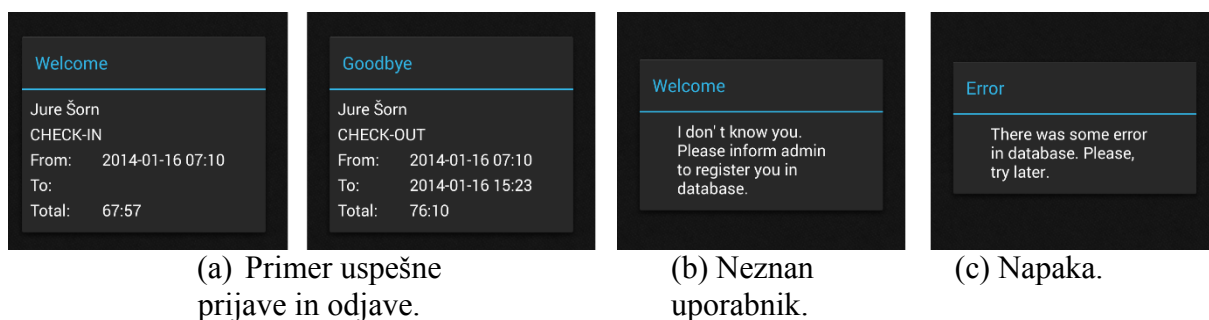
Aplikacija NFC Signer se mora, ko odkrije, da se je približala značka NFC, ustrezno odzvati. Za ustrezen odziv poskrbi razred `MainActivity`. Značke NFC so lahko naprave različnih tipov, zato za kategorizacijo nastavimo filter v datoteki `AndroidManifest.xml`. Ko je sistem za prioriteto odpremo značk nastavljen, je aktivnost pripravljena za branje značk NFC. Ob približanju značke se ustvari namen, za katerega skrbi funkcija `onNewIntent`, ki kliče `resolveIntent` (Slika 4.6). Omenjena funkcija prebere `EXTRA_ID`, ki je identifikator značke NFC. Sprva je ID v zlogovnem zapisu (angl. *byte*), nato pa ga s pomočjo funkcije `Long.toString()` in metode `getLong()` spremenimo v besedilo.

```
private void resolveIntent(Intent intent) {
    String action = intent.getAction();
    if (NfcAdapter.ACTION_TAG_DISCOVERED.equals(action)||
        NfcAdapter.ACTION_TECH_DISCOVERED.equals(action)||
        NfcAdapter.ACTION_NDEF_DISCOVERED.equals(action)) {
        byte[] id = intent.getBytesExtra(NfcAdapter.EXTRA_ID);
        eventAPI=nfcSignerDB.returnNFCInfo(Long.toString(getLong(id)));
        if(eventAPI.isError()){
            UserResultsDialog userDialog=new UserResultsDialog(this,true);
            userDialog.showHands();
        }
        else if(eventAPI.isUserKnown()){
            UserResultsDialog userDialog=new UserResultsDialog(this,eventAPI);
            userDialog.showHands();
        }
        else{
            UserResultsDialog userDialog=new UserResultsDialog(this,false);
            userDialog.showHands();
        }
    }
}
```

Slika 4.6: Funkcija `resolveIntent`, ki z značke NFC odčita `EXTRA_ID` in izvede določeno akcijo.

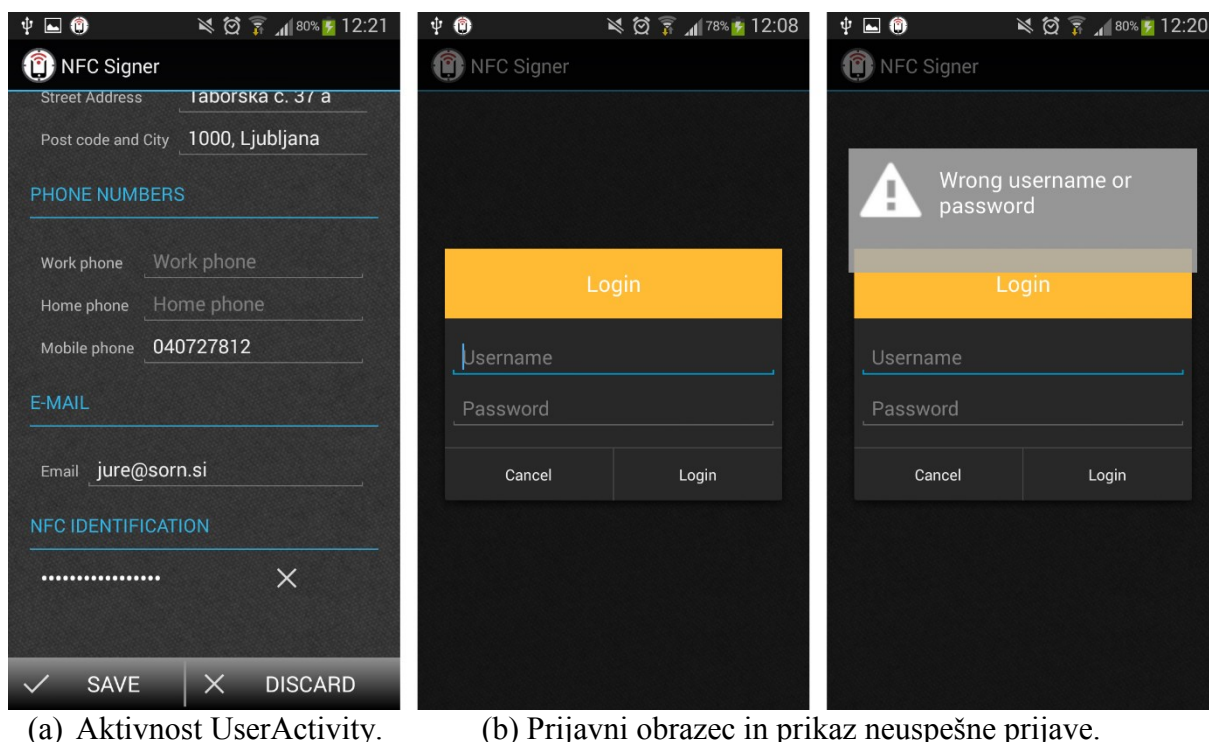
Na podlagi prebrane vrednosti se s pomočjo razreda `DataBaseAdapter` identifikator primerja z že shranjenimi in prikaže se eno izmed dialoških oken. Če identifikator pripada znanemu uporabniku, se dogodek shrani v podatkovno bazo in prikaže se dialog, ki vsebuje čas prihoda, čas odhoda in skupno število ur, ki jih je uporabnik naredil v mesecu (vidno na Sliki 4.7). Kadar v podatkovni bazi pride do napake ali v primeru, ko identifikator ne pripada

znanemu uporabniku, se pojavi temu ustrezno dialoško okno. Pojavi se za štiri sekunde in nato izgine.



Slika 4.7: Dialoška okna ob približanju kartice NFC.

Podobna programska koda za odkrivanje značk NFC je uporabljena tudi pri aktivnosti `UserActivity`, le da se v tem primeru identifikator NFC shrani v lokalno podatkovno bazo s pomočjo razreda `DataBaseAdapter`. Aktivnost opravlja delo shranjevanja in dodajanja uporabnikov (Slika 4.8a). Pravico za dodajanje uporabnikov ima zgolj skrbnik. V primeru dodajanja in urejanja se prikaže obrazec za prijavo (Slika 4.8b). Po uspešni prijavi je mogoče izvajanje želenih operacij.



Slika 4.8: Prikaz aktivnosti `UserActivity` in prijave uporabnika.

4.2.4 Razvoj in implementacija spletnega strežnika na mobilni napravi

Po zaključku razvoja in implementacije podatkovne baze ter identifikacije kartic NFC je bilo treba razviti spletni strežnik. Spletni strežnik zažene aktivnost MainActivity kot storitev (angl. *service*); to pomeni, da se izvaja kot proces v ozadju in je neodvisen od aktivnosti ter deluje tudi po tem, ko smo aplikacijo že zapustili. Storitve je treba implementirati v datoteki AndroidManifest.xml.

Spletni strežnik smo v celoti razvili s pomočjo odprtokodnega projekta Jetty. Jetty poskrbi za delovanje http-strežnika in http-klienta ter podpira javax servlete. Ker je sprogramiran v javanskem programskem jeziku, je primeren za želene naloge v operacijskem sistemu Android. Do sedaj je bilo razvitih že devet verzij projekta Jetty. Izbrali smo verzijo 7.x, saj deluje na enaki verziji Java kot izbrana minimalna verzija Androida. Spletni strežnik Jetty posreduje spletne strani prek vrat HTTP 8080 ter vrat HTTPS (angl. *HyperText Transfer Protocol Secure*) 8443, ki so zavarovana različica HTTP in za šifriranje prometa pred vmesnimi opazovalci uporabljajo SSL (angl. *Secure Sockets Layer*) (Slika 4.9) [21].

```
protected void configureConnectors()
{
    if (server != null)
    {
        SelectChannelConnector nioConnector = new SelectChannelConnector();
        nioConnector.setUseDirectBuffers(false);
        nioConnector.setPort(_port);
        server.addConnector(nioConnector);
        Log.i(TAG, "Configured "+SelectChannelConnector.class.getName()+" on port "+_port);
        SslContextFactory sslContextFactory = new SslContextFactory();
        sslContextFactory.setKeyStore(_keystoreFile);
        sslContextFactory.setTrustStore(_truststoreFile);
        sslContextFactory.setKeyStorePassword(_keystorePassword);
        sslContextFactory.setKeyManagerPassword(_keymgrPassword);
        sslContextFactory.setKeyStoreType("bks");
        sslContextFactory.setTrustStorePassword(_truststorePassword);
        sslContextFactory.setTrustStoreType("bks");
        SslSelectChannelConnector sslConnector = new SslSelectChannelConnector(sslContextFactory);
        sslConnector.setPort(_sslPort);
        server.addConnector(sslConnector);
        Log.i(TAG, "Configured "+sslConnector.getClass().getName()+" on port "+_sslPort);
    }
}
```

Slika 4.9: Implementacija dveh konektorjev (angl. *connector*), ki skrbita za povezavo po protokolih HTTP in HTTPS preko določenih vrat.

Dostop do spletne aplikacije imajo zgolj prijavljeni uporabniki. Jetty ima za namene avtentikacije kar nekaj razredov. Za nas je bil najbolj primeren FormAuthenticator, ki s pomočjo prirejenega razreda AnroidMappedLoginService uporabniku dodeli veljavno sejo. AnroidMappedLoginService v lokalni podatkovni bazi preveri, ali se uporabniško ime in

geslo ujemata. V primeru ujemanja skrbnike in ostale uporabnike preusmeri na določen naslov, v primeru neujemanja pa na naslov za nepravilno prijavo.

Za delo z lokalno podatkovno bazo preko spletne aplikacije smo razvili pet servletov HTTP, funkcije katerih so opisane v Tabeli 4.3.

Servlet HTTP	Naslov	Funkcije	Vrsta vsebine
AdminServlet	/admin/api/*	pregled trenutnega stanja uporabnika	application/json
		dodajanje in urejanje uporabnika	
		dodajanje in urejanje dogodka	
		brisanje dogodka in uporabnika	
SignerLogServlet	/api/getMyHours/*	pregled dogodkov uporabnika	application/json
UserServlet	/api/user/*	urejanje uporabniškega imena in gesla	application/json
		preverjanje prostega uporabniškega imena	
		pridobivanje imena in priimka uporabnika	
DownloadServlet	/admin/download/*	prenos datoteke Excel	application/vnd.ms-excel
LogoutServlet	/api/logout/*	razveljavitev seje in preusmeritev naslova	/

Tabela 4.3: Prikaz in opis vseh funkcij servletov HTTP.

Strežniški programi (angl. *servlet*) na zahtevo odjemalca posredujejo podatke preko protokola HTTP. Izpostavili bi strežniški program na naslovu /admin/download, ki dostopa do lokalne podatkovne baze in podatke posreduje v tabelarnem formatu Excelove datoteke. Tabela vsebuje 5 stolpcev (dan, skupno število delovnih ur v dnevu, čas prihoda, čas odhoda, tip dogodka) (Slika 4.10). V datoteki so tudi ime osebe ter leto in mesec evidence. Odjemalcu, torej uslužbencu, ki vodi evidentiranje delovnega časa, z možnostjo izvoza datoteke olajšamo delo. S preprostimi funkcijami programa Microsoft Office Excel lahko kasneje izračuna želene postavke, kot je na primer izračun plač.

	A	B	C	D	E
1	PersonName	Janez Novak			
2	Year	2015		Month	2
3					
4					
5	Day	Total	Check In	Check Out	Type
6	01.02.2015	0:00	-	-	-
7	02.02.2015	7:00	2015-02-02 08:38	2015-02-02 15:38	Regular
8	03.02.2015	8:38	2015-02-03 08:00	2015-02-03 16:38	Regular
9	04.02.2015	7:37	2015-02-04 08:01	2015-02-04 15:38	Regular
10	05.02.2015	6:37	2015-02-05 09:01	2015-02-05 15:38	Regular
11	06.02.2015	8:00	2015-02-06 08:00	2015-02-06 16:00	Holiday
12	07.02.2015	0:00	-	-	-
13	08.02.2015	0:00	-	-	-
14	09.02.2015	7:00	2015-02-09 08:00	2015-02-09 15:00	Regular
15	10.02.2015	7:29	2015-02-10 08:58	2015-02-10 16:27	Regular
16	11.02.2015	7:12	2015-02-11 09:00	2015-02-11 16:12	Regular
17	12.02.2015	8:06	2015-02-12 08:06	2015-02-12 16:12	Regular
18	13.02.2015	8:06	2015-02-13 08:06	2015-02-13 16:12	Regular
19	14.02.2015	0:00	-	-	-
20	15.02.2015	0:00	-	-	-
21	16.02.2015	8:09	2015-02-16 07:06	2015-02-16 15:15	Regular
22	17.02.2015	8:54	2015-02-17 07:18	2015-02-17 16:12	Regular
23	18.02.2015	7:36	2015-02-18 08:24	2015-02-18 16:00	Regular
24	19.02.2015	9:15	2015-02-19 07:12	2015-02-19 16:27	Regular
25	20.02.2015	8:00	2015-02-20 08:00	2015-02-20 16:00	Official Absence
26	21.02.2015	0:00	-	-	-
27	22.02.2015	0:00	-	-	-
28	23.02.2015	7:33	2015-02-23 06:54	2015-02-23 14:27	Regular
29	24.02.2015	7:33	2015-02-24 07:54	2015-02-24 15:27	Regular
30	25.02.2015	8:24	2015-02-25 07:03	2015-02-25 15:27	Regular
31	26.02.2015	8:48	2015-02-26 07:39	2015-02-26 16:27	Regular
32	27.02.2015	8:00	2015-02-27 08:00	2015-02-27 16:00	Regular
33	28.02.2015	0:00	-	-	-
34					
35	Total	157:57			

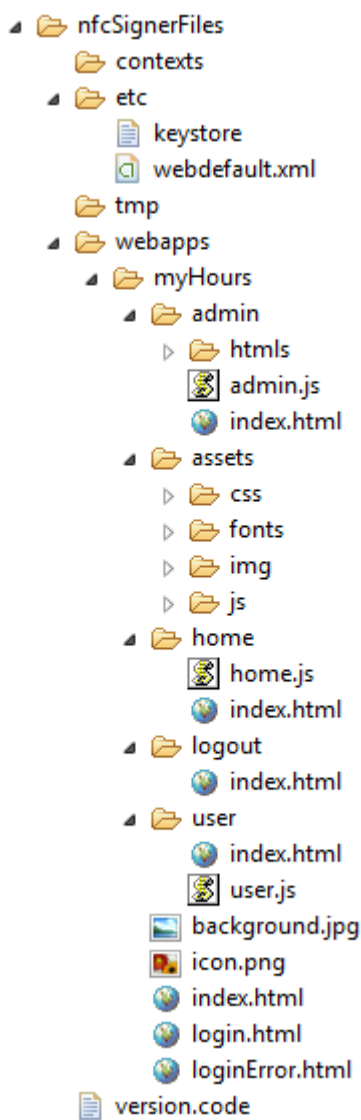
Slika 4.10: Izvoz podatkov v obliki datoteke Excel.

4.2.5 Razvoj spletne aplikacije

Celotna spletna aplikacija je bila razvita v urejevalniku besedil Sublime Text 2.

4.2.5.1 Struktura spletne aplikacije

Spletni strežnik Jetty za prepoznavanje datotek spletne aplikacije uporablja določena pravila in ogrodja. V mapi etc se hranijo vse konfiguracijske datoteke, ki upravljajo nastavitve spletnega strežnika, in datoteke za delovanje varne povezave. Mapa tmp shranjuje začasne datoteke [22]. V mapo contexts je mogoče dodati neobvezne izbirne konfiguracijske datoteke in datoteke, ki se pojavijo na začetnem naslovu [23]. V mapi webapps so shranjene vse datoteke spletne aplikacije (Slika 4.11). Celotno ogrodje nfcSignerFiles se pri prvi uporabi mobilne aplikacije prekopira v pomnilnik mobilnega telefona.



Slika 4.11: Struktura ogrodja Jetty.

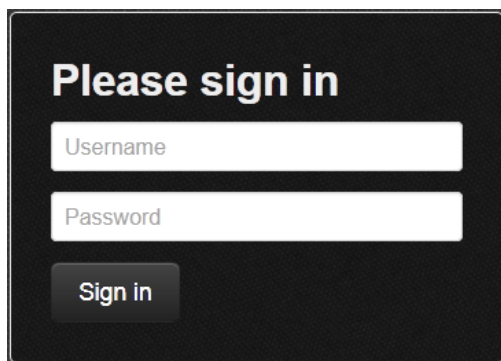
V vsako mapo spletne aplikacije je mogoče dodati tudi dodatne konfiguracije spletne aplikacije, ki jih vnaprej spišemo v programski kodi.

Spletno aplikacijo smo realizirali s pomočjo orodij HTML, CSS, JavaScript in Bootstrap.

4.2.5.2 Zasloni spletne aplikacije

Spletna stran loči uporabnike na skrbnike in zaposlene. Uporabnik z vlogo skrbnika ima pregled nad trenutnim stanjem zaposlenih in njihovimi delovnih urami ter ima pravico do urejanja preteklih dogodkov ter dodeljevanja dopustov in bolniških odsotnosti. Zaposleni ima pregled nad svojimi delovnimi urami ter dovoljenje za urejanje svojega profila. Uporabnik se v spletno aplikacijo prijavi na spletni strani.

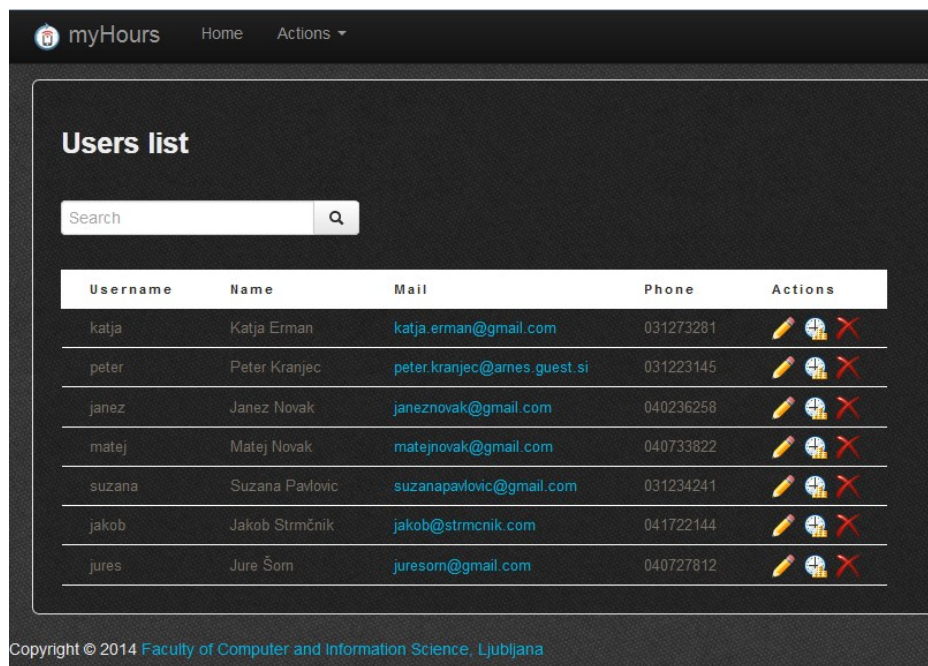
Uporabnik se v spletno aplikacijo prijavi na strani za prijavo, ki ga preusmeri na ustrezen naslov (Slika 4.12).

The image shows a login interface with a dark background. At the top, the text "Please sign in" is displayed in a bold, white font. Below this, there are two white input fields. The first field is labeled "Username" in a light gray font. The second field is labeled "Password" in a light gray font. Below the password field, there is a dark gray button with the text "Sign in" in white.

Slika 4.12: Obrazec za prijavo.

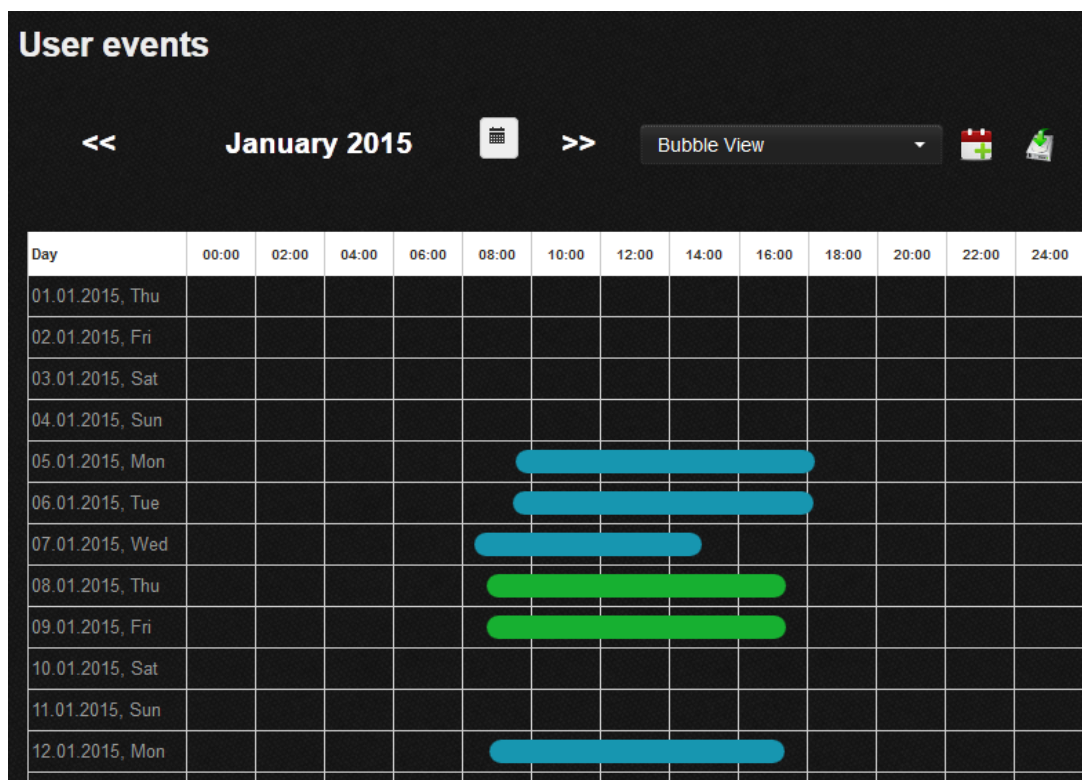
Skrbnik

Skrbnik lahko na spletni strani vseh uporabnikov pregleda celoten seznam uporabnikov, poišče specifičnega uporabnika in ga po želji izbriše, uredi njegove podatke ali pa pregleda njegovo prisotnost na delovnem mestu (Slika 4.13).



Slika 4.13: Seznam zaposlenih.

Dogodki so prikazani za obdobje enega meseca (Slika 4.14). Skrbnik ima možnost pregleda dogodkov v tabelarnem ali ploščinskem načinu, dodajanja novih dogodkov in izvoza datoteke tipa .xls z vsebino mesečne evidence zaposlenega. Vsak tip dogodka je druge barve, prav tako pa lahko za vsak ustvarjen dogodek skrbnik s klikom na dogodek odpre podrobnosti dogodka. To mu prikaže obrazec, enak obrazcu za dodajanje dogodka (Slika 4.15).



Slika 4.14: Seznam dogodkov.

The screenshot shows the 'Event details' form for 06.01.2015. The form includes fields for 'Check in' (2015-01-06 08:43) and 'Check out' (2015-01-06 16:46), both with clock icons. The 'Type' is set to 'Regular'. Below these fields, there are labels for 'Day Hours' and 'Event Hours', both showing '8:03'. A 'Save' button is at the bottom.

Slika 4.15: Podrobnosti dogodka.

Zaposlena oseba

Zaposlena oseba lahko na svoji spletni strani pregleda svojo evidenco delovnih ur, ki je skoraj enaka kot na spletni strani skrbnika, le brez možnosti dodajanja novih dogodkov. Zaposleni

lahko pregleda podrobnosti vsakega dogodka, kar je (izvzemši možnost urejanja) vidno na Sliki 4.14.

4.3 Razvoj strežnika za potrebe sinhronizacije

Vsaka naprava ima svojo življenjsko dobo. Poleg tega nanjo vpliva še vrsta zunanjih dejavnikov, kot so mehanske poškodbe, kraje in splošne okvare ali odpovedi delovanja. Kot opisano v preteklih poglavjih, se vsi podatki mobilne naprave hranijo v lokalni podatkovni bazi. Zaradi potrebe po varnem kopiranju podatkov vseh mobilnih aplikacij smo razvili spletni servis, ki svoje podatke hrani v podatkovni bazi MySQL. Dostop do spletnega servisa je omogočen le registriranim uporabnikom. V nadaljevanju bosta predstavljena podatkovni model, ki je ključnega pomena za implementacijo sinhronizacije, in potek komunikacije med mobilno napravo in strežnikom.

4.3.1 Izdelava podatkovnega modela

Za potrebe sinhronizacije podatkov smo definirali entitete **users**, **events** in **companies**, ki hranijo vse podatke podjetij. Entiteti **users** in **events** sta enaki kot v mobilni aplikaciji, dodali smo le identifikator podjetja. Entiteta **companies** skrbi za shrambo vseh podjetij, ki želijo uporabljati varnostno kopiranje svojih podatkov na mobilni aplikaciji.

Entiteti **users** in **events** sta že bili predstavljeni v razdelku 4.2.1, zato bodo v nadaljevanju opisani zgolj atributi entitete **companies** (Tabela 4.4).

Polje	Opis polja
id	Šifra podjetja. Identifikator podjetja.
username	Uporabniško ime. Uporabniško ime podjetje potrebuje za izvedbo spletne storitve sinhronizacije.
password	Geslo uporabnika. Uporabniško geslo za izvedbo spletne storitve sinhronizacije.
name	Ime podjetja.
street	Ulica in hišna številka podjetja.
city	Mesto podjetja.

country	Država podjetja.
----------------	------------------

Tabela 4.4: Opis strukture podatkov entitete **companies**.

4.3.2 Spletni servis

Razvoj spletnega servisa se začne pri izbiri spletnega strežnika. Izbrali smo JBOSS EAP (angl. *JBoss Enterprise Application Platform*), saj omogoča hitro vzpostavitev s podatkovno bazo MySQL. Povezavo z MySQL na strežniku JBoss vzpostavimo zgolj z dodanimi nastavitvami v konfiguracijski datoteki `standalone.xml` ter dodanim gonilnikom JDBC (angl. *Java database connectivity*) [26]. Tak način povezave omogoča kasnejše upravljanje s podatkovno bazo MySQL s pomočjo objektnega jezika, saj lahko navaden javanski razred implementiramo kot entiteto. Razvoj spletnega servisa je v celoti potekal v razvojnem okolju JBoss Developer Studio, ki je namenjen za delo s strežnikom Jboss ter za razvoj spletnih aplikacij JBoss. Jboss Developer Studio je osnovan na razvojnem okolju Eclipse.

Najpomembnejšo vlogo pri varnem kopiranju podatkov ima centralni strežnik JBoss, ki nosi odgovornost za hranjenje podatkov o podjetjih, njihovih zaposlenih in njihovih podatkih evidentiranja. Naloge varnega kopiranja opravlja spletni servis s podatkovno bazo, ki je javno dostopen na internetu preko protokola HTTP. Za implementacijo spletnega servisa je bil izbran arhitekturni stil REST (angl. *Representational State Transfer*).

S pomočjo spletnega servisa in podatkovne baze MySQL so na strežniku izpostavljene štiri spletne storitve na pripadajočih naslovih:

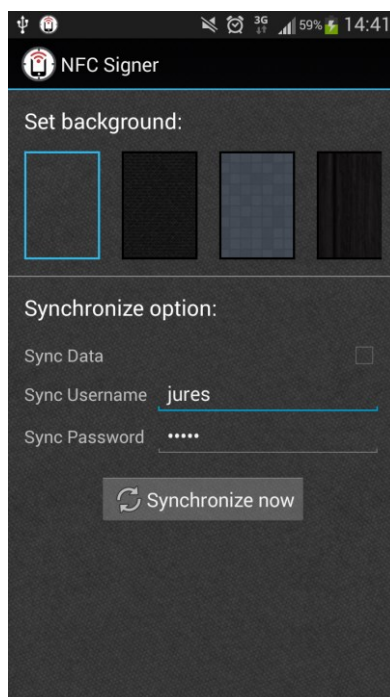
- **/rest/get/count**: Storitev vsebuje preprosto poizvedbo v podatkovni bazi, ki vrne število dogodkov in zaposlenih nekega podjetja.
- **/rest/get/events**: Poslovna logika storitve vsebuje preprosto poizvedbo, ki vrne vse podatke dogodkov. Podatki se prenašajo v obliki JSON.
- **/rest/get/users**: Poslovna logika storitve vsebuje preprosto poizvedbo, ki vrne vse podatke uporabnikov. Podatki se prenašajo v obliki JSON.
- **/rest/post/data**: Storitev skrbi za posodabljanje dogodkov in uporabnikov v podatkovni bazi MySQL.

Te štiri storitve nam omogočajo, da lahko hitro in učinkovito v celoti sinhroniziramo podatke med podatkovno bazo MySQL in lokalnimi bazami mobilnih naprav. Vse te storitve imajo

omejen dostop in so na voljo le prijavljenim uporabnikom. V ta namen je uporabljena t. i. form avtentikacija. Avtentikacija odjemalcu dodeli identifikator seje, 16-bajtni naključno dolg niz znakov, zgoščen z algoritmom MD5 [27]. V primeru pravilne prijave seja postane veljavna in uporabnik lahko dostopa do spletnih storitev.

4.3.3 Sinhronizacija na strani mobilne naprave

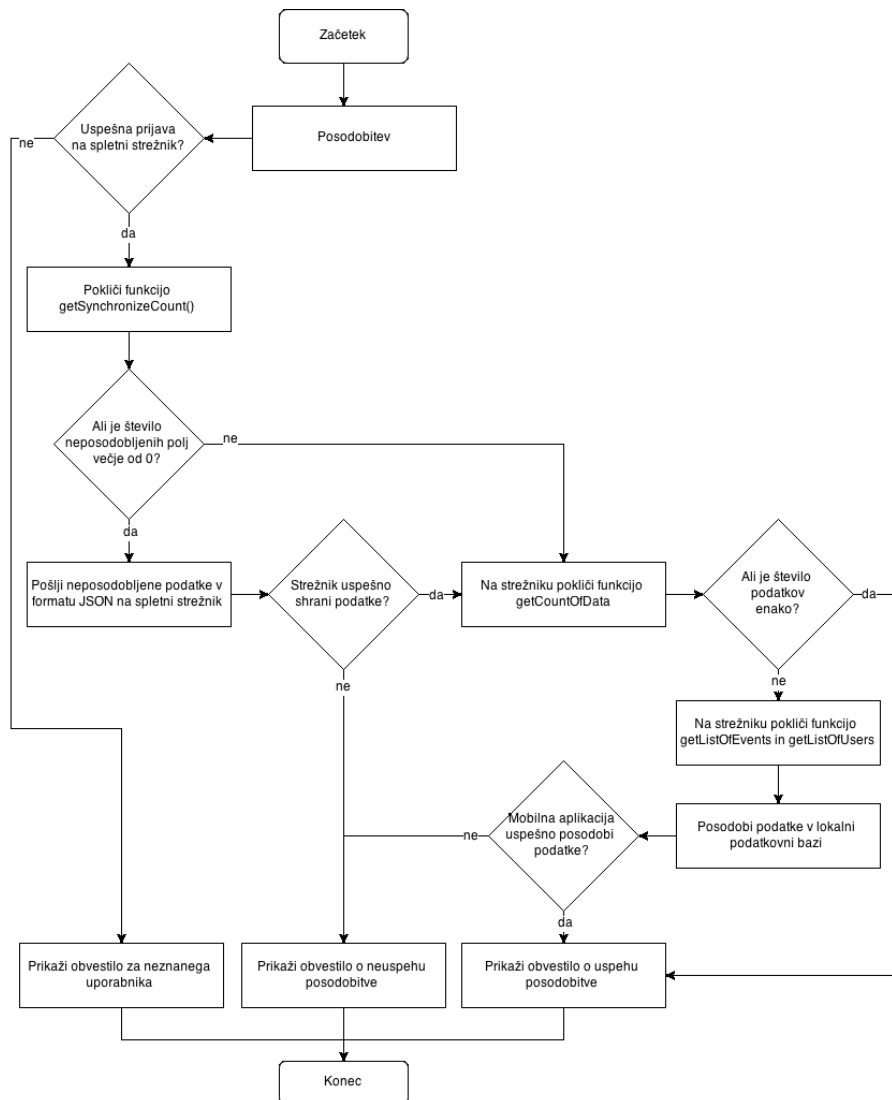
Vsako podjetje, ki uporablja mobilno aplikacijo NFC Signer, ima možnost sinhronizacije svojih podatkov, s čimer naredi varnostno kopijo. Vkllop ali izkllop možnosti je dovoljen le skrbniškim uporabniškim računom, ki se prijavijo s prijavnim obrazcem, prikazanim na Sliki 4.8b v razdelku 4.2.3. Po vklopu sinhronizacije se posodabljanje vrši avtomatsko vsako uro in ob vsakem prvem zagonu aplikacije. Posodabljanje je možno izvesti tudi ročno s pritiskom na gumb v nastavitvah, kot je prikazano na Sliki 4.16.



Slika 4.16: Prikaz zaslonske maske v aktivnosti nastavitvev, kjer lahko s klikom na gumb Synchronize now ročno posodobimo podatke.

Če je na voljo internetna povezava, aplikacija preveri spremembe podatkov. To stori s pregledom parametra **isSynchronize** vsakega polja entitete **users** in **events** v lokalni podatkovni bazi. Podatke nato pošlje na strežnik v obliki JSON, kjer se podatki posodobijo. Ob uspešni posodobitvi podatkov na centralnem strežniku s pomočjo storitve REST, ki je izpostavljena na naslovu `/rest/get/count`, pridobimo število podatkov na strani strežnika. V primeru enakega števila se posodobitev zaključi z obvestilom o uspehu, v nasprotnem primeru

pa vse podatke pridobimo preko storitve REST na naslovu `/rest/get/event` in `/rest/get/users` ter jih posodobimo v lokalni podatkovni bazi. Na Sliki 4.14 je prikazan diagram poteka posodobitve podatkov.



Slika 4.17: Diagram poteka posodobitve na mobilni napravi.

4.4 Analiza delovanja sistema

Na trgu obstajajo različni sistemi za spremljanje prihodov in odhodov. NFC Signer je mobilna rešitev za evidentiranje prisotnosti s pametnimi telefoni z omogočeno tehnologijo NFC za zaposlene, ki delo opravljajo na različnih lokacijah na terenu ali na stacionarnih lokacijah. V mobilno rešitev je vključena spletna aplikacija, do katere je moč dostopati z brskalnikom in uporabo brezžičnega usmerjevalnika ali z omogočeno prenosno dostopno točko (angl. *hot*

spot). Omogočena prenosna dostopna točka na mobilni napravi služi za povezavo preko omrežja Wi-Fi, na katerega se povežemo z računalnikom ali drugimi napravami. Tako lahko mobilna naprava z nameščeno aplikacijo nudi dostop do spletnih strani, ki omogočajo urejanje podatkov o vseh zaposlenih ter njihovih prihodih in odhodih. Za uporabo vseh funkcij sistema dostop do interneta ni potreben, saj mobilna naprava sama služi kot brezžični oddajnik, identifikacijski terminal in strežnik. V primeru uporabe brezžičnega usmerjevalnika je razlika ta, da povezavo med računalnikom in mobilno napravo prevzame usmerjevalnik.

V času testiranja smo ugotovili, da spletni strežnik, ki teče na mobilni napravi, ne dela težav. Omejitve predstavlja napajanje mobilne naprave; če želimo stalno delujoč spletni strežnik, je treba mobilno napravo polniti. Rešitev za ta problem bi bila postavitve naprave na neko dostopno točko, kjer bi bila naprava stalno priklopljena na električno omrežje ali pa bi delo polnjenja mobilne naprave prevzel vodja delovne ekipe. Pomanjkljivost spletnega strežnika na napravi je tudi pokritost podjetja. Če bi podjetje želelo pokriti več vhodov z mobilnimi napravami, bi bilo bolje uporabiti en sam centralni strežnik. Uporaba centralnega strežnika ima določene prednosti; tako pri posodobitvi spletne aplikacije posodobitev mobilne aplikacije ne bi bila potrebna. Prav tako je verjetnost, da se zaradi različnih mehanskih poškodb in ostalih vzrokov pokvari strežnik, manjša. Dostop do spletne aplikacije ima uporabnik preko internetne povezave, kar pomeni, da je neodvisen od delovanja mobilne aplikacije na napravi.

Priporočena je dodatna zaščita dostopa do nastavitve mobilne naprave s tovrstnimi aplikacijami, saj bi lahko zlonamernež z lahkoto izbrisal systemske in druge datoteke ter s tem napravo praktično uničil. Obstaja tudi nevarnost za namerno poškodbo in krajo naprave ali namerno motenje delovanja sistema za evidentiranjem z neprestanim klicanjem naprave. Reševanje teh težav smo prepustili uporabnikom samim. Rešitev bi lahko predstavljali kamera na dostopni točki in enostavna blokada klicev.

Tehnologija NFC se je v sistemu NFC Signer izkazala za učinkovito. Testna mobilna naprava z delujočo aplikacijo odčita identifikator značk NFC v razdalji okoli 3 centimetrov. Odziv na odčitano značko NFC in shranjevanje prihoda ali odhoda zaposlenih je hitrejši od sekunde.

Poglavje 5 Sklepne ugotovitve

V diplomski nalogi smo predstavili rešitev, ki na delovnem mestu omogoča hitro in preprosto identifikacijo zaposlenih. Za identifikacijo zaposlenih smo uporabili tehnologijo NFC, ki je učinkovita in preprosta za uporabo. Prepričali smo se, da je tehnologija NFC primerna za uporabo v sistemih za registracijo časa. Koncept rešitve NFC Signer omogoča uporabo mobilne naprave kot mobilnega terminala za evidentiranje delovnega časa, medtem ko se za identifikatorje uporablja značke NFC in ostale mobilne naprave s podporo za tehnologijo NFC. Programsko rešitev lahko uporablja vsako podjetje, ki si želi cenovno ugodno rešitev za evidentiranje svojih zaposlenih. Potreben je le nakup mobilne naprave z operacijskim sistemom Android 4.3, ki ima podporo za tehnologijo NFC. Za delovanje spletnega vmesnika poskrbi naprava sama z omogočenim oddaljenim dostopom (angl. *hot spot*). Rešitev je naklonjena tudi uporabi, kjer klasični registracijski terminali ne pridejo v poštev, kot je na primer menjava lokacije dela. V tem primeru je vodja delovne ekipe opremljen s pametnim mobilnim telefonom, ki omogoča uporabo tehnologije NFC in ima naložen sistem NFC Signer, ostali zaposleni pa se enostavno evidentirajo s svojimi prenosnimi telefoni ali karticami NFC.

Nadaljnji razvoj sistema vidimo v kadrovske podpori podjetja in analitiki. Uporabnik bi lahko zaprosil za dopust preko spletnega vmesnika, delodajalec pa bi ga lahko kasneje odobril. Za spremljanje prisotnosti je v sistemu poskrbljeno, vendar bi lahko v primeru zamude ali neprihoda zaposlenega na delovno mesto mobilna aplikacija sama obvestila delodajalca. V sistem bi lahko dodali tudi plačne rešitve, ki bi bile namenjene obračunu plač in ostalih prejemkov. Mobilni del rešitve je moč dopolniti s funkcijami, ki bi uporabniku dovolile izbiro dogodka, ki ga želi registrirati.

Mobilna aplikacija vsebuje funkcije, ki podjetju omogočajo korektno vodenje evidence zaposlenih. Vodilni imajo vpogled v trenutno stanje zaposlenih ter možnost urejanja in dodajanja rednih registracij uslužbencev, bolniške odsotnosti, dopustov in službenih odsotnosti z dela. Prav tako je poskrbljeno za varnost podatkov na napravi, saj aplikacija omogoča sinhronizacijo le-teh. S tem smo izpolnili zahtevane cilje, ki smo si jih zastavili v načrtovanju projekta.

Prihodnost nam prinaša vedno večje število pametnih telefonov z vgrajeno tehnologijo NFC, zato bo njena uporaba vse bolj primerna. Tehnologija NFC ustreza zahtevam za identifikacijo zaposlenih, saj je povezava varna, čas vzpostavitve pa hiter.

Literatura

- [1] (2015) Wikipedia: Near Field Communication. Dosegljivo: http://sl.wikipedia.org/wiki/Near_Field_Communication.
- [2] (2015) NFC: History of Near Field Communication. Dosegljivo: <http://www.nearfieldcommunication.org/history-nfc.html>.
- [3] Vedat Coskun, Kerem Ok, Busra Ozdenizci, NFC Application Development for Android, Wiley, 2012, str. 23–51, 115–205.
- [4] NFC Forum, NFC Data Exchange Format (NDEF), Technical Specification, Verzija 1.0, julij 2006.
- [5] NFC Forum, Record Type Definition (RTD), Technical Specification, Verzija 1.0, julij 2006.
- [6] (2015) Vedat Coskun, Kerem Ok, Busra Ozdenizci, Near Field Communication: From Theory to Practice, Wiley, 2012.
- [7] (2015) Microsoft: Inside NFC: Usages and working Principle. Dosegljivo: http://developer.nokia.com/community/wiki/Inside_NFC:_Usages_and_Working_Principles.
- [8] John Wiley & Sons, Inc., »Professional Android 4 Application Development«, Indiana, 2012, str. 693–699.
- [9] (2015) Wikipedia: Android (operating_system). Dosegljivo: http://en.wikipedia.org/wiki/Android_operating_system.
- [10] (2015) SQLite. Dosegljivo: <http://www.sqlite.org/about.html>.
- [11] (2015) Jetty (web server). Dosegljivo: [http://en.wikipedia.org/wiki/Jetty_\(web_server\)](http://en.wikipedia.org/wiki/Jetty_(web_server)).
- [12] (2015) JSON. Dosegljivo: <http://en.wikipedia.org/wiki/JSON>.

- [13] (2015) MySQL. Dosegljivo: <http://dev.mysql.com/doc/refman/4.1/en/what-is-mysql.html>.
- [14] (2015) RedHat JBoss EAP. Dosegljivo: https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.4/pdf/Development_Guide/JBoss_Enterprise_Application_Platform-6.4-Development_Guide-en-US.pdf.
- [15] (2015) Apache Maven Project. Dosegljivo: <http://maven.apache.org/>
- [16] (2015) Samsung I9300 Galaxy S III. Dosegljivo: http://www.gsmarena.com/samsung_i9300_galaxy_s_iii-4238.php.
- [17] (2015) Samsung I9500 Galaxy S4. Dosegljivo: http://www.gsmarena.com/samsung_i9500_galaxy_s4-5125.php.
- [18] Splošni pogoji poslovanja sistema Tušmobil. Dosegljivo: http://www.tusmobil.si/uploads/docs/STORITVE/Poslovne_storitve/NFC_SPP_V1.pdf.
- [19] Adamsoft. Evidenca delovnega časa. Dosegljivo: <http://www.adamsoft.si/TinA/>.
- [20] (2015) Registracija delovnega časa Četrta pot. Dosegljivo: <http://www.cetrta.net/registracija-delovnega-casa/evidenca-delovnega-casa.html>.
- [21] (2015) Secure Sockets Layer. Dosegljivo: http://sl.wikipedia.org/wiki/Secure_Sockets_Layer.
- [22] (2015) Temporary Directories. Dosegljivo: <http://eclipse.org/jetty/documentation/current/ref-temporary-directories.html>.
- [23] (2015) Jetty/Howto/Deploy Web Applications. Dosegljivo: https://wiki.eclipse.org/Jetty/Howto/Deploy_Web_Applications.
- [24] (2015) Sublime Text. Dosegljivo: <http://www.sublimetext.com/>.
- [25] (2015) Bootstrap. Dosegljivo: <http://getbootstrap.com/components/>.
- [26] (2015) Using Connector/J with JBoss. Dosegljivo: <http://dev.mysql.com/doc/connector-j/en/connector-j-usagenotes-jboss.html>
- [27] (2015) Form-based Authentication. Dosegljivo: <https://access.redhat.com/documentation/en->

US/JBoss_Enterprise_Application_Platform/5/html/Security_Guide/Enabling_FORM
_Authentication.html.